Selfi: Self-improving Reconstruction Engine via 3D Geometric Feature Alignment

Youming Deng[1,2] Songyou Peng[2] Junyi Zhang[2,3] Kathryn Heal[2]
Tiancheng Sun[2] John Flynn[2] Steve Marschner[1] Lucy Chai[2]
[1]Cornell University [2]Google [3]UC Berkeley

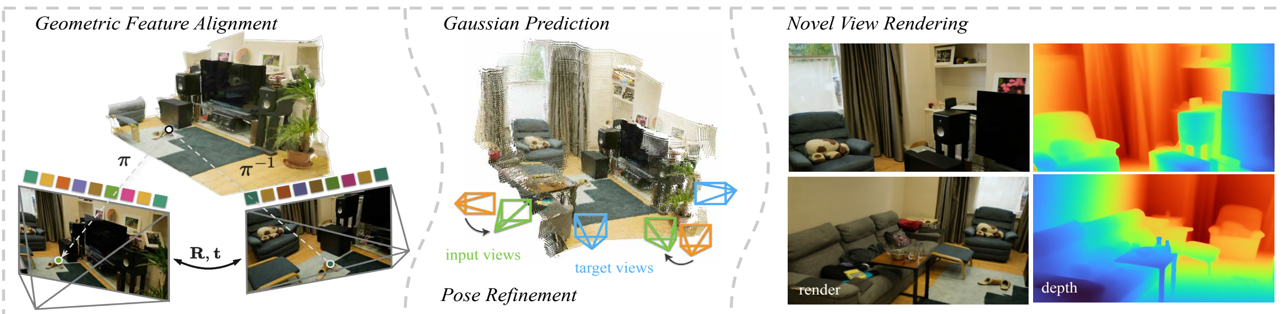


Figure 1. **Self-improving Reconstruction Engine.** We introduce *Selfi*, a self-improving pipeline for novel view synthesis from unposed images. We start by learning geometrically aligned features using consistency losses and self-labelled pseudo ground truths from a 3D foundation model (*e.g.,* VGGT [54]). These features can be used to predict Gaussian primitives [23], and also refine initial poses via bundle adjustment. The improved poses are used to further adjust the initial 3D representation, resulting in an even higher quality final rendering.

Abstract

Novel View Synthesis (NVS) has traditionally relied on models with explicit 3D inductive biases combined with known camera parameters from Structure-from-Motion (SfM) beforehand. Recent vision foundation models like VGGT take an orthogonal approach – 3D knowledge is gained implicitly through training data and loss objectives, enabling feed-forward prediction of both camera parameters and 3D representations directly from a set of uncalibrated images. While flexible, VGGT features lack explicit multi-view geometric consistency, and we find that improving such 3D feature consistency benefits both NVS and pose estimation tasks. We introduce Selfi, a self-improving 3D reconstruction pipeline via feature alignment, transforming a VGGT backbone into a high-fidelity 3D reconstruction engine by leveraging its own outputs as pseudo-ground-truth. Specifically, we train a lightweight feature adapter using a reprojection-based consistency loss, which distills VGGT outputs into a new geometrically-aligned feature space that captures spatial proximity in 3D. This enables state-of-the-art performance in both NVS and camera pose estimation, demonstrating the benefits of feature alignment for downstream 3D reasoning. More details on our project page: `https://denghilbert.github.io/selfi`

1. Introduction

Novel view synthesis (NVS) has long relied on known camera parameters or those recovered from an SfM-first pipeline: detect keypoints, match them, and solve for cameras before optimizing a scene representation for rendering [23, 30, 34, 40, 41]. While effective, this decoupling between the cameras and scene representation is not only computationally intensive but also fragile – NVS quality is highly dependent on the accuracy of SfM poses. Feed-forward NVS methods remove per-scene optimization and move toward direct prediction. Given calibrated images, they extract image features and lift them to 3D primitives or pixels in one forward pass [6, 11, 12, 20, 21, 45, 56, 63, 66, 71, 79, 80]. However, most approaches still assume known cameras from SfM, so the quality also degrades when calibration is inaccurate or even fails without SfM estimation.

The advent of 3D Vision Foundation Models (VFMs) [25, 54, 59] has offered a paradigm shift. Trained on vast, diverse datasets with 3D annotations, these models can predict camera poses, dense depth, and 3D structure from uncalibrated images in a single forward pass, effectively bypassing SfM. A promising recent direction is to leverage these VFMs for NVS by directly decoding VFM features into 3D representations like 3D Gaussians [19, 70]. However, this approach suffers from a significant drawback: the resulting NVS quality is substantially lower than that of optimization-based methods. We hypothesize this is because VFM features, while powerful for the geometric prediction tasks they were trained on, are not explicitly optimized to be geometrically consistent across different views, which is crucial for high-fidelity NVS.

This paper overcomes that limitation: we propose a surprisingly simple feature alignment strategy to transform a pre-trained 3D VFM into a state-of-the-art NVS and pose estimation engine *without using any 3D ground-truth annotations*. Our key insight is to leverage the VFM's own outputs as a dense, self-supervised signal to learn a new, geometrically-aligned feature space. We freeze VGGT as a backbone foundation model and train a lightweight feature adapter by constructing a self-supervised task. Using pseudo-ground-truth depth and cameras from VGGT, we reproject query points from one view to other views to serve as a correspondence signal. The feature adapter head produces per-pixel features for both images; we then enforce a simple reprojection-based feature consistency loss between features at their corresponding locations. This yields geometrically aligned features where feature similarity captures both semantic content and proximity in 3D space without any camera annotations or 3D supervision beyond the model outputs themselves.

The resulting features are powerful and versatile. First, when used to predict parameters for 3D Gaussian Splatting [23], our learned features achieve state-of-the-art NVS quality from unposed images, dramatically outperforming methods that use the original VGGT features. Second, these geometrically-aligned features help establish robust correspondences, allowing us to achieve better performance on pose estimation with a few extra bundle adjustment (BA) steps. Both of these are achieved in a fully self-supervised manner. In summary, our contributions are:

- **Self-improving geometric feature learning from unposed RGB**. We introduce a reprojection-based feature consistency loss to learn geometrically aligned 3D features, without any annotations.
- **State-of-the-art NVS and pose estimation results with a frozen backbone**. Despite training only small heads on top of a frozen VFM, we set a new SOTA on unposed NVS and pose estimation benchmarks.

2. Related Work

3D Reconstruction and Pose Estimation. Earlier 3D reconstruction pipelines rely on decoupled two-stage processes, with Structure-from-Motion (SfM) for geometry and cameras followed by Multi-view Stereo (MVS) for depth [8, 22, 34, 38–41, 69]. While these methods are grounded in multi-view geometry [16], they are slow and lack robustness. More recent learning-based approaches use diffusion models [52, 74, 78] and transformers [50, 53, 55] to directly infer geometric quantities like camera pose, depth, and 3D point maps in an end-to-end manner [25, 32, 51, 57–60, 67, 73]. These 3D Vision Foundation Models (VFMs), trained on large, diverse datasets, have demonstrated impressive generalization capabilities. Our method builds on VGGT [54], a VFM designed to jointly predict all necessary geometric quantities, which we use as our geometric pseudo-label teacher.

Feed-Forward Novel View Synthesis. NVS aims to generate realistic images from arbitrary viewpoints. One class of NVS methods, such as those built on NeRF [30], 3D Gaussian Splatting (3DGS) [23], or voxels [14, 31, 44], optimize a representation individually per scene and typically require calibrated inputs from SfM or a fixed camera rig [4, 12, 26, 65]. In contrast, generalizable NVS approaches train a network to directly regress a 3D scene representation (e.g., volumetric fields or Gaussian parameters) from one or more input images in a feed-forward fashion, but many still assume known camera parameters. These methods may incorporate inductive biases like cost volumes [6, 11], depths [64], epipolar constraints [43, 56], or multi-plane images [11, 13, 49, 79]. More recently some methods directly feed the inputs into a general-purpose transformer architecture [66, 75], resulting in a renderable representation [5, 45–47, 71]. Variations of this task aim to reconstruct entire scenes in a single inference pass rather than localized viewing angles [17, 80]. Given the recent rise in models that also predict camera poses or pointmaps in a feed-forward fashion, several works jointly regress Gaussian splat parameters with cameras and geometry, allowing for NVS from unposed images [19, 70, 77]. An orthogonal approach taken by LVSM [20] and Rayzer [18] avoids a 3D representation entirely, and instead directly outputs the rendered image. Our approach follows the per-pixel Gaussian splat parameterization; we train a lightweight adapter on top of the features derived from VGGT that yields an explicit 3D scene representation for rendering to new camera viewpoints.

Vision Foundation Models as a Feature Backbone. Feature representations learned from large-scale VFMs like DINO, CLIP, and more [9, 33, 35, 42] have proven invaluable for various downstream 3D tasks, including correspondence matching and depth estimation [2, 25, 54, 59, 68]. Prior works have also leveraged diffusion features for semantic correspondences [15, 29, 48, 72], while Feat2GS [7] probes the representation space of various visual foundation models using NVS as a proxy task for 3D understanding. However, these features are primarily semantic and lack the dense geometric consistency required for high-fidelity 3D reconstruction. Rather than simply reusing existing VFM features, we use features from VGGT to learn a geometri-

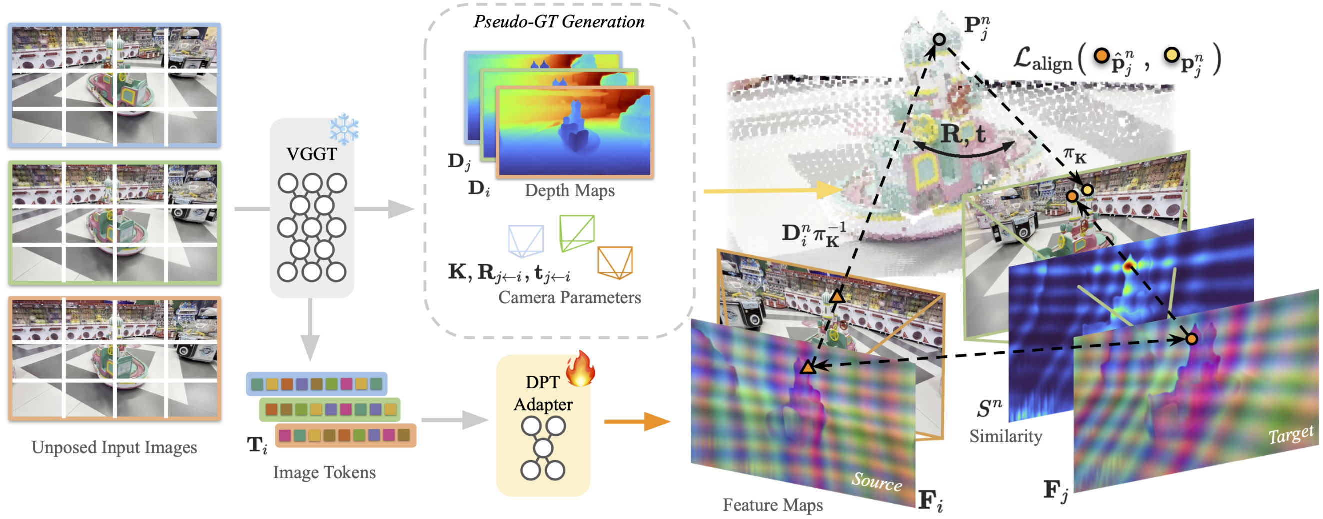


Figure 2. **Geometric Feature Alignment with Self-Labeled Pseudo-Ground-Truth**. Using a pretrained VGGT [54] backbone, we use predicted depth and camera parameters as pseudo-ground-truth to align features obtained from a DPT adapter on top of VGGT image tokens. We sample query points and reproject these points to a target view using depth and camera parameters. Our loss function encourages the features at these two corresponding locations from source and target frames to be similar.

cally aligned representation space, and we demonstrate that these features not only benefit NVS, but can be used to refine predicted poses and achieve even better rendering.

3. Method

In this section, we first introduce our training strategy for geometric feature alignment to obtain a powerful 3D-aware representation in Sec. 3.1. Next, we feed these features into an efficient 3D Gaussian predictor for Novel View Synthesis (NVS) in Sec. 3.2. Finally, we leverage our features for matching and dense bundle adjustment (BA) to further improve pose estimation and novel view synthesis in Sec. 3.3.

3.1. Geometric Feature Alignment

While NVS methods with explicit 3D inductive biases [6, 12, 13, 23, 30, 71] are able to maintain geometric and appearance consistency when rendering to novel views by design, this property is not guaranteed in transformer-based foundation models. Since 3D VFMs like DUSt3R [59] and VGGT [54] learn 3D priors implicitly, their feature spaces are not natively multi-view consistent. To bridge this gap, we propose learning a geometrically aligned feature space as a prerequisite for high-fidelity NVS [19, 28, 70].

Feature Correspondence Prediction. To produce our aligned features, we take the feature backbone and aggregator from pre-trained VGGT [54] and append a spatial feature adapter DPT [36]:

$$\mathbf{F}_i = \text{DPT}_{\text{adapter}}(\mathbf{T}_i), \tag{1}$$

where image tokens $\mathbf{T}_i \in \mathbb{R}^{4\times(H_p\times W_p)\times D}$ are taken from four intermediate feature maps from the VGGT backbone with H_p, W_p, and D as the feature spatial resolution and feature dimension. $\mathbf{F}_i \in \mathbb{R}^{H\times W\times C}$ is the output feature map for an arbitrary frame i, with H and W the same as the output image resolution, and $C = 24$.

The goal of our feature alignment is to train the feature adapter such that features corresponding to spatially proximal 3D locations exhibit high similarity. Given a sampled query location n, source frame s, and target frame t, we take the pixel-aligned feature $\mathbf{F}_s^n \in \mathbb{R}^C$ from the 2D source feature map $\mathbf{F}_s$, and the 2D target feature map $\mathbf{F}_t$. We compute a similarity map in Eq. (2) and apply a softmax with temperature τ in Eq. (3):

$$S^n(u,v) = \frac{\mathbf{F}_s^n \cdot \mathbf{F}_t(u,v)}{\|\mathbf{F}_s^n\|_2 \, \|\mathbf{F}_t(u,v)\|_2}, \tag{2}$$

$$w^n(u,v) = \frac{\exp(S^n(u,v)/\tau)}{\sum_{u',v'} \exp(S^n(u',v')/\tau)}. \tag{3}$$

The predicted 2D correspondence $\hat{\mathbf{p}}_t^n$ (shown as an orange dot • in Fig. 2) is obtained as a weighted average over the (u,v) target coordinates.

$$\hat{\mathbf{p}}_t^n = \sum_{u,v} w^n(u,v)\,[u,v], \tag{4}$$

We find that using a weighted average over all pixels enables the model to leverage information from all target features during alignment, providing denser supervision compared to contrastive training methods [25].

Pseudo-Ground-Truth Supervision from VGGT. To supervise the predicted correspondence, we establish pseudo-ground-truth 2D-2D correspondences using VGGT. Specifically, given a set of images, we run VGGT [54] to obtain per-frame depth maps $\mathbf{D}_i$, shared camera intrinsics $\mathbf{K}$, and relative poses $[\mathbf{R}_{j\leftarrow i}|\mathbf{t}_{j\leftarrow i}]$. For a query pixel $\mathbf{p}_s^n$ in source frame s, we compute its corresponding location in a target frame t via 3D unprojection and reprojection. We unproject the pixel using its depth $\mathbf{D}_s^n$ to 3D space (orange triangle ▲ in Fig. 2), transform it to the target coordinate system, and project it back to 2D to obtain the pseudo-GT correspondence $\mathbf{p}_t^n$ (yellow dot ● in Fig. 2):

$$\mathbf{P}_t^n = \mathbf{R}_{t\leftarrow s}\mathbf{D}_s^n\pi_{\mathbf{K}}^{-1}\mathbf{p}_s^n + \mathbf{t}_{t\leftarrow s}, \tag{5}$$

$$\mathbf{p}_t^n = \pi_{\mathbf{K}}\mathbf{P}_t^n, \tag{6}$$

where $\pi_{\mathbf{K}}$ denotes the projection from 3D camera coordinates to the 2D pixel space with intrinsic $\mathbf{K}$, and $\pi_{\mathbf{K}}^{-1}$ denotes the inverse projection. To handle possible occlusions during reprojection, we also maintain a hard visibility map $\mathbf{V}_t^n$, filtering by the difference between the z-coordinate of the 3D point backprojected from the source view, and the target-view depth map at the corresponding re-projected location (which we refer to as $\mathbf{D}_t^n$, the value of $\mathbf{D}_t$ at pixel location $\mathbf{p}_t^n$):

$$\mathbf{V}_t^n = \left[\left|\mathbf{P}_t^n \cdot [0\ 0\ 1]^T - \mathbf{D}_t^n\right| < \alpha\right], \tag{7}$$

where $[\cdot]$ denotes the Iverson bracket.

Objective for Alignment. Given our predicted correspondence $\hat{\mathbf{p}}_t^n$ and reprojection-based pseudo-ground truth $\mathbf{p}_t^n$, we optimize the features to produce high similarity at the corresponding points, weighted by visibility:

$$\mathcal{L}_{\text{align}} = \frac{1}{TN}\sum_{t=1}^{T}\sum_{n=1}^{N}\mathbf{V}_t^n\left\|\hat{\mathbf{p}}_t^n - \mathbf{p}_t^n\right\|_2^2. \tag{8}$$

We demonstrate in our baseline experiments Sec. 4.2 and ablations Sec. 4.4 that our alignment strategy benefits downstream NVS performance. Moreover, we show that our aligned features can further refine the initial camera poses from VGGT via bundle adjustment, despite training on pseudo-GT with zero projection error in Sec. 4.3.

3.2. Feed-Forward Gaussian Prediction

Using the aligned feature maps from the previous step, we now predict 3D Gaussian parameters [23] to enable feed-forward novel view rendering. We freeze the $\text{DPT}_{\text{adapter}}$ feature head and train a new U-Net [37] decoder. This U-Net takes the aligned feature $\mathbf{F}_s$ concatenated with the source input image I_s, and predicts the parameters for each Gaussian primitive relative to the source camera coordinates. Specifically, the decoder outputs: quaternions $\mathbf{q}_s$, scales $\mathbf{s}_s$, color $\mathbf{c}_s$, opacity $\boldsymbol{\sigma}_s$, and a depth residual $\Delta\mathbf{D}_s$ to be added to the initial depth map $\mathbf{D}_s$:

$$\mathbf{F}_s^{\text{dec}} = \text{U-Net}(\text{cat}(\mathbf{F}_s, I_s)), \tag{9}$$

$$\mathbf{q}_s, \Delta\mathbf{D}_s = \text{Conv}_q(\mathbf{F}_s^{\text{dec}}), \text{Conv}_{\mathbf{D}}(\mathbf{F}_s^{\text{dec}}), \tag{10}$$

$$\{\boldsymbol{\sigma}_s, \mathbf{s}_s, \mathbf{c}_s\} = \text{MLP}(\mathbf{F}_s^{\text{dec}}), \tag{11}$$

For the 3D positions of the Gaussians, we add the predicted depth residual to the initial depth map and unproject the 2D pixels into 3D locations $\boldsymbol{\mu}_s$ relative to the source images' coordinate frames:

$$\boldsymbol{\mu}_s = (\mathbf{D}_s + \Delta\mathbf{D}_s)\pi_{\mathbf{K}}^{-1}\mathbf{p}_s. \tag{12}$$

We predict colors $\mathbf{c}_s$ using spherical harmonics coefficients to model view-dependent effects [14, 23]. One key modification in our work is that we also enable spherical harmonics on the density $\boldsymbol{\sigma}_s$ [17], rather than using a single scalar density. As the geometry prediction from VGGT may not be fully accurate, in particular in low-confidence regions, we find this view-dependent density crucial for overcoming possible occlusions and misalignments. In effect, the density spherical harmonics serves as a learned confidence metric; for any given render viewpoint, it modulates opacity so that unreliable Gaussians become transparent. This also enables us to prune the low-confidence Gaussians for rendering efficiency, a distinct approach from the voxelization pruning used by AnySplat [19] and WorldMirror [28]. We quantify this design choice in our ablation (Sec. 4.4 and Tab. 6) and visualize the learned densities in Fig. 6.

Finally, the collection of 3D Gaussian parameters produced from all source images s are then rasterized to obtain target renderings $\hat{I}_t$ using the predicted relative camera parameters $\mathbf{C}_{t\leftarrow s}$. The model is trained solely with an RGB reconstruction loss:

$$\hat{I}_t = \text{Rasterizer}(\{(\boldsymbol{\sigma}_s, \mathbf{s}_s, \mathbf{c}_s, \boldsymbol{\mu}_s, \mathbf{q}_s, \mathbf{C}_{t\leftarrow s})\}_{s=1}^{S}), \tag{13}$$

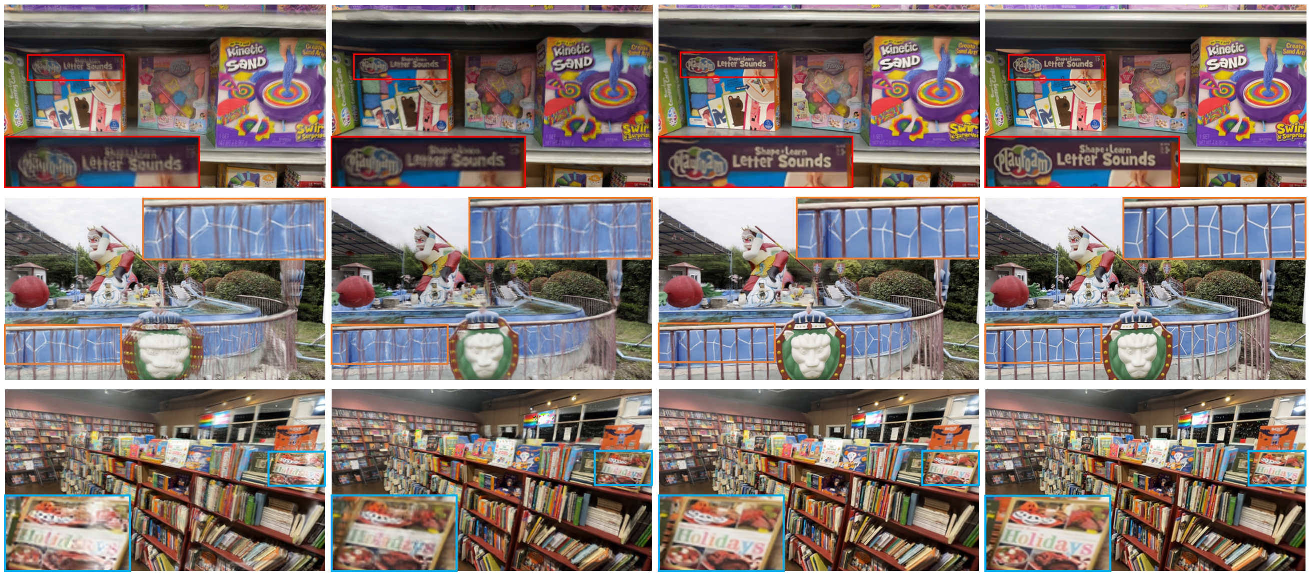
$$\mathcal{L}_{\text{RGB}} = \frac{1}{T}\sum_{t=1}^{T}\left\|\hat{I}_t - I_t\right\|. \tag{14}$$

3.3. Dense Bundle Adjustment with Depth Shift

Compared to other feed-forward 3DGS methods [19, 28, 70] that perform post-optimization on both camera parameters and 3D Gaussians, our aligned features offer a more classic and efficient alternative to improve camera parameters with a quickly-converging bundle adjustment (BA).

While BA using correspondences from our aligned features improves the estimated poses (Sec. 4.3), it also changes the position of sparse 3D points associated with 2D correspondences. This requires us to also adjust the Gaussian centers to be consistent with this new geometry. We find that, because the gradient from BA may change the estimated intrinsics, the 3D points will move along each camera's z-axis (in the depth direction) to compensate. Ignoring this change in depth before rasterizing the Gaussian primitives leads to rendering misalignment shown in Fig. 4a.

We observe that the change in depth is primarily a linear function, as visualized in Fig. 4c. Therefore, it can be easily modeled with an affine transformation $\phi(\cdot)$ estimated from the sparse BA points and applied to all densely predicted



(a) AnySplat [19] (b) WorldMirror [28] (c) **Ours** (d) GT

Figure 3. **Qualitative Comparisons on DL3DV [27].** We visualize novel view renderings from AnySplat [19], WorldMirror [28], and our method. Our method successfully recovers thin structures, such as guardrails, and fine-grained details, such as the text "Holidays".

Gaussians. We apply $\phi(\cdot)$ to the per-frame depth maps $\mathbf{D}_s + \Delta\mathbf{D}_s$, with Gaussian scales adjusted accordingly:

$$\boldsymbol{\mu}'_s = \phi(\mathbf{D}_s + \Delta\mathbf{D}_s)\pi^{-1}_{\mathbf{K}'}\mathbf{p}_s \tag{15}$$

$$\mathbf{s}'_s = \frac{\phi(\mathbf{D}_s + \Delta\mathbf{D}_s)}{\mathbf{D}_s + \Delta\mathbf{D}_s} \cdot \mathbf{s}_s. \tag{16}$$

where $\boldsymbol{\mu}'_s$ is the new 3D position scaled by the new affine-corrected depth and $\mathbf{K}'$ is the new intrinsics. The scale $\mathbf{s}'_s$ is adjusted proportionally. As shown in Fig. 4b and in the experiments, this simple correction successfully bridges the geometric gap, allowing for pose improvements from BA and enhancing NVS quality.

(a) w/o Depth Shift (b) w/ Depth Shift (c) Shift Function

Figure 4. **Bundle Adjustment with Depth Shift.** (a) After refining the camera poses with bundle adjustment, naively rendering the predicted Gaussian primitives with the new poses results in misalignment. (b) Propagating the adjustments in sparse 3D points during BA to the dense depth maps results in improved rendering. (c) We plot the sparse point depths before and after BA, and observe that a linear fit suffices for this adjustment.

4. Experiments

In this section, we first briefly introduce the datasets used for training, implementation details, and evaluation metrics in Sec. 4.1. We demonstrate the performance of our method by comparing it with various novel view synthesis (NVS) baselines in Sec. 4.2, and subsequently show in Sec. 4.3 that our aligned features are effective for bundle adjustment (BA) and can further improve pose estimation. Finally, we present ablation studies on all the design choices described in Sec. 4.4.

4.1. Experimental Setup

Datasets. We train and test our model on two large-scale public datasets: DL3DV [27] and RealEstate10K [79]. Both datasets contain diverse indoor and outdoor real-world scenes at different scales, enabling the model to be robust and generalizable across various scenes. We also include conventional datasets like MipNeRF [62] and Tanks&Temples [24] for NVS evaluation.

Training View Sampling. For geometric feature alignment, we sample 11 frames with different strides. The middle frame serves as the source frame, and all others are used as targets. This stage is trained exclusively on DL3DV. For the Gaussian U-Net decoder, we combine both DL3DV and RealEstate10K. We sample 6 source frames at various strides, and additionally, 5 frames between the source frames are used as the target views. More details are in the supplementary.

Implementation Details. We train the geometric features using a DPT [36] decoder with the AdamW optimizer for

Dataset	Method	Short (6 Frames)			Medium (12 Frames)			Long (24 Frames)			Longer (36 Frames)		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
DL3DV [27]	3DGS [23]	25.63	0.8376	0.1985	27.92	0.8794	0.1678	27.97	0.8776	0.1745	28.47	0.8875	0.1693
	AnySplat [19]	18.84	0.5665	0.2949	18.16	0.5297	0.3323	17.41	0.4871	0.3769	17.25	0.4727	0.3969
	WorldMirror [28]	21.76	0.7389	0.2162	20.79	0.6884	0.2620	19.56	0.6067	0.3243	19.24	0.6007	0.3412
	Ours	**24.94**	**0.8442**	**0.1566**	**22.96**	**0.7849**	**0.2090**	**21.58**	**0.7302**	**0.2509**	**19.97**	**0.6632**	**0.3119**
RealEstate10K [79]	3DGS [23]	28.46	0.9034	0.1477	29.57	0.9221	0.1275	30.72	0.9361	0.1025	28.47	0.8875	0.1693
	AnySplat [19]	23.88	0.7949	0.1836	21.49	0.7439	0.2352	19.99	0.6932	0.2727	21.38	0.7344	0.2377
	WorldMirror [28]	25.54	0.8691	0.1502	23.99	0.8464	0.1759	22.21	0.7883	0.2103	24.48	0.8521	0.1642
	Ours	**28.34**	**0.9021**	**0.1206**	**27.46**	**0.8974**	**0.1317**	**26.27**	**0.8815**	**0.1469**	**25.37**	**0.8537**	**0.1602**

Table 1. **Novel View Synthesis with Varying Sequence Length**. We compare our method against several baselines on the RealEstate10K [79] and DL3DV [27] datasets. As the number of input frames increases, the performance of all feed-forward methods degrades, as it becomes more challenging to predict consistent 3D Gaussians from a greater number of views. In contrast, 3DGS [23] with GT camera parameters, which we include as an upper bound, improves with more views as it can better optimize for consistency.

(a) AnySplat [19] (b) WorldMirror [28] (c) **Ours** (d) GT

Figure 5. **Qualitative Comparisons on RealEstate10K [79]**. We visualize novel view renderings from AnySplat [19], WorldMirror [28], and our model. Our method more faithfully reconstructs details such as the door hinge and the tiled wall.

150K iterations. We use a cosine learning rate scheduler with a peak learning rate of 1×10^{-4} and a warm-up phase of 1K steps. We randomly sample 4,096 query points on the source frame for training. The geometric feature alignment step takes roughly 2 days on 128 NVIDIA H100 GPUs. Next, to train the Gaussian head, we use the same optimizer and number of iterations but with a higher learning rate of 2×10^{-4}. The input and target rendering resolution is fixed to 294×518. This stage uses 128 NVIDIA H100 GPUs and takes about 1.5 days. The visibility threshold α is set to 0.05, and the softmax temperature τ is 100. The entire project is implemented in JAX [3]. More details are in the supplementary.

4.2. NVS with Feed-forward Gaussians

Using our lightweight Gaussian head on top of geometrically aligned features, we outperform relevant baselines by a large margin in rendering quality. Despite using uncalibrated inputs, our method achieves similar or superior performance when compared to other methods that assume known poses, including pixelNeRF [71], GPNR [43], Du et al. [10], PixelSplat [5], MVSplat [6], DepthSplat [64], GS-LRM [75], Long-LRM [80], LVSM [20], and ReSplat [63]. At the same time, we outperform existing pose-free feed-forward Gaussian methods such as NoPoSplat [70], Flare [77], AnySplat [19], and WorldMirror [28] by a significant margin. We also include per-scene optimized 3DGS [23] with ground-truth camera parameters as the strongest baseline for comparison.

For comprehensive evaluation, we evaluate performance on: 1) Varying number of input views; 2) Varying degrees of overlap among input views, by changing the sampling stride of the images; 3) Two-view evaluation split from PixelSplat [5] on RealEstate10K [79]. All NVS evaluations are conducted on hold-out scenes from both RealEstate10K [79] and DL3DV [27].

Comparisons over Varying Sequence Lengths. We use a fixed minimal sampling stride (5 for RealEstate10K, 2 for DL3DV) with varying numbers of input images (6, 12, 24, and 36), corresponding to longer sequences. This setup tests the robustness of NVS models to different trajectory lengths while maintaining a relatively consistent coverage density. As detailed in Tab. 1, our method consistently outperforms all baselines across all settings, and is even comparable to the per-scene optimization-based 3DGS with GT camera parameters for shorter sequences. We do not report the performance of Flare [77] in Tab. 1 due to its limitations

Dataset	Method	Small (Sparse)			Medium (Regular)			Large (Dense)			Average		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
DL3DV [27]	3DGS [23]	19.58	0.6628	0.3517	25.55	0.8365	0.2000	25.63	0.8376	0.1985	23.59	0.7790	0.2501
	AnySplat [19]	11.97	0.3251	0.5173	14.01	0.3845	0.4447	18.84	0.5665	0.2949	14.94	0.4254	0.4189
	Flare [77]	13.86	0.3223	0.5145	15.26	0.3715	0.4281	18.93	0.5364	0.2429	16.02	0.4101	0.3952
	WorldMirror [28]	14.98	0.4641	0.4363	17.03	0.5500	0.3571	21.76	0.7389	0.2162	17.92	0.5843	0.3365
	Ours	**19.46**	**0.7288**	**0.2589**	**22.52**	**0.8057**	**0.1903**	**24.94**	**0.8442**	**0.1566**	**22.30**	**0.7929**	**0.2019**
RealEstate10K [79]	3DGS [23]	21.79	0.7755	0.2695	24.40	0.8405	0.2160	28.46	0.9034	0.1477	24.88	0.8398	0.2111
	AnySplat [19]	18.02	0.6427	0.2993	20.09	0.6993	0.2554	23.88	0.7949	0.1836	20.66	0.7123	0.2461
	Flare [77]	20.12	0.6423	0.2092	21.66	0.6950	0.1624	24.25	0.7767	**0.1089**	22.01	0.7047	0.1602
	WorldMirror [28]	**21.31**	0.8005	0.2163	22.65	0.8283	0.1924	25.54	0.8691	0.1502	23.17	0.8326	0.1863
	Ours	21.13	**0.8575**	**0.1609**	**23.95**	**0.8835**	**0.1390**	**28.34**	**0.9021**	0.1206	**24.47**	**0.8810**	**0.1402**

Table 2. **Novel View Synthesis with Varying Overlap**. We vary the degree of overlap between the input images by changing the sampling stride. With more overlap, all methods demonstrate improved rendering performance, with our method achieving the best performance overall compared to other feed-forward baselines. Notably, our method can achieve performance on par with 3DGS [23], despite 3DGS using ground-truth poses and SfM point initialization.

	Method	PSNR ↑	SSIM ↑	LPIPS ↓
Pose-Required	pixelNeRF [71]	20.43	0.589	0.550
	GPNR [43]	24.11	0.793	0.255
	Du *et al.* [10]	24.78	0.820	0.213
	pixelSplat [5]	26.09	0.863	0.136
	MVSplat [6]	26.39	0.869	0.128
	DepthSplat [64]	27.47	0.889	0.114
	GS-LRM [75]	28.10	0.892	0.114
	Long-LRM [80]	28.54	0.895	0.109
	LVSM (enc-dec) [20]	28.58	0.893	0.114
	LVSM (dec-only) [20]	29.67	0.906	0.098
	ReSplat [63]	**29.72**	0.911	0.100
Pose-Free	NoPoSplat [70]	26.82	0.880	0.125
	Flare [77]	26.91	0.873	0.127
	Ours	29.01	**0.942**	**0.053**

Table 3. **Quantitative Comparison with the Two-View Convention**. We follow the two-view convention previously used by PixelSplat [5] on RealEstate10K [79]. Except for Flare [77], NoPoSplat [70], and our method, all other methods require calibrated images as input. Our method remains competitive even against those that require ground-truth camera parameters.

on the number of inputs.

Comparisons over Varying Degrees of Overlap. We use a fixed number of input images (6 frames) with varying sampling strides, where larger strides correspond to smaller overlap (*i.e.,* sparse-view NVS). We test strides of 5, 10, and 15 on RealEstate10K [79], and 2, 4, and 8 on DL3DV [27]. As shown in Tab. 2, our method consistently outperforms all pose-free baselines, with performance approaching or exceeding 3DGS [23] in challenging sparse-view settings. This demonstrates the effectiveness of our feature alignment objectives in creating a robust and generalized 3D scene representation, even when geometric input is sparse.

Qualitative Comparison. Our NVS results shown in Fig. 3 and Fig. 5, using six input frames with the large overlap setting, are noticeably sharper and recover higher-frequency details, like text and thin structures, than the baseline methods, further validating the effectiveness of our method.

Two-View Evaluation. We compare our method against a range of pose-required two-view and sparse-view reconstruction methods on 7k pairs from RealEstate10K [79] in Tab. 3. Our method achieves the best results for SSIM [61] and LPIPS [76] among all compared methods (both pose-required and pose-free baselines). We hypothesize that the slightly lower PSNR for our model can be attributed to exposure differences in the two inputs and the fact that our model is trained for multi-view NVS, so its robustness is slightly reduced when inputs are limited to only two views.

Dataset	Bundle Adjustment	Novel View Synthesis			Pose Estimation		
		PSNR↑	SSIM↑	LPIPS↓	AUC@3	AUC@5	AUC@15
DL3DV [27]	✗	24.94	0.844	0.156	0.813	0.876	0.955
	✓	25.13	0.850	0.152	0.870	0.916	0.970
MipNeRF360 [1]	✗	23.27	0.730	0.212	0.261	0.394	0.711
	✓	23.46	0.741	0.204	0.312	0.454	0.775
Tanks&Temples [24]	✗	19.41	0.746	0.200	0.818	0.886	0.961
	✓	19.54	0.755	0.198	0.955	0.973	0.991

Table 4. **BA for Joint Pose Estimation and NVS.** Using the aligned features, we refine the camera poses using BA and further adjust the predicted Gaussian positions to be consistent with the BA output. This self-refinement operation yields further improvements over the initial NVS without BA, which we demonstrate on DL3DV and apply zero-shot to MipNeRF360 and Tanks&Temples.

Evaluation on the RayZer [18] Split. We evaluate our method on RayZer setup and will further illustrate in the supplementary.

4.3. BA for Pose Estimation and NVS

We evaluate pose refinement by running BA on VGGT's initial predictions using correspondences from both Co-Tracker [22] and our method (see Tab. 5). Our feature-based matching yields superior pose accuracy across varying input sizes. Crucially, our method scales robustly to larger input sets (*e.g.,* > 40 images) where Co-Tracker fails due to memory exhaustion.

We further validate that this improvement in pose jointly improves NVS. We take these refined camera poses and sparse Gaussian locations, move all predicted Gaussians

Method	AUC@3	AUC@5	AUC@15	AUC@30
10 frames				
VGGT w/o BA [54]	0.7900	0.8600	0.9470	0.9680
VGGT w/ BA [22]	0.8350	0.8840	0.9277	0.9734
Ours	**0.8670**	**0.9142**	**0.9690**	**0.9842**
40 frames				
VGGT w/o BA [54]	0.8583	0.9014	0.9602	0.9787
VGGT w/ BA [22]	0.8720	0.9137	0.9588	0.9824
Ours	**0.8819**	**0.9235**	**0.9713**	**0.9846**
100 frames				
VGGT w/o BA [54]	0.8447	0.8977	0.9617	0.9796
VGGT w/ BA [22]	Out-of-Memory			
Ours	**0.8762**	**0.9192**	**0.9699**	**0.9839**

Table 5. **Pose Estimation Evaluation**. VGGT achieves reasonable performance, but the estimated poses can be further improved via BA. We report results for different numbers of input images and compare our BA results against Co-Tracker [22]. Our method consistently improves the predictions, even when Co-Tracker [22] fails due to out-of-memory.

with the affine depth correction following Sec. 3.3, and obtain further improvements in NVS, as shown in Tab. 4 on DL3DV [27], MipNeRF360 [1], and Tanks&Temples [24].

4.4. Ablation Study

Geometric Feature Alignment. We observe that our geometrically aligned features prove crucial for obtaining a high-quality Gaussian head. To ablate this, we directly adopt features from the tracking head of VGGT, with an identical training schedule. The first two rows of Tab. 6 show that our features, despite also being trained independently from the Gaussian head, offer a strong improvement in NVS metrics. The importance of feature alignment is also corroborated in our comparison to WorldMirror [28], which directly passes feature tokens to the Gaussian head.

Spherical Harmonics (SH) Prediction. We ablate the importance of predicting SHs for both RGB color and density in Tab. 6. Unlike the original 3DGS [23], which assigns only a scalar density to each Gaussian, density SHs play a crucial role for NVS in helping to combat noisy estimations in pose and depth, as shown in rows 4 and 5 of Tab. 6. We provide visualizations in Fig. 6 for better understanding. The first row shows six input images, with the target camera placed at their midpoint. For each input view, we obtain a set of Gaussians and rasterize them into the target camera view. We show the rasterized results without (second row) and with (last row) view-dependent density in Fig. 6. Our model learns to use Gaussians from the closest input frames, while Gaussians from inputs farther from the target are assigned nearly zero density. We interpret this behavior as a learned confidence score over density, and find that it greatly improves rendering quality.

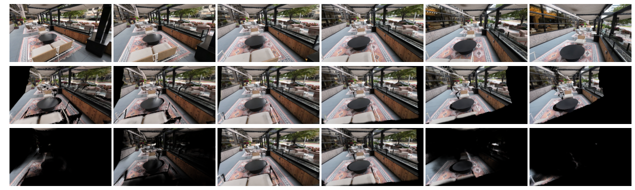


Figure 6. **View-dependent Density.** Given six input views (top row), we render the Gaussians from each individual input to a target camera at the midpoint of the two center views. This process is shown for models trained without (middle row) and with (bottom row) view-dependent density. The view-dependent density serves as a confidence score that learns to downweight input views that are farther from the target view.

Feature Alignment	Density SHs	RGB SHs	Bundle Adjustment	Depth Shift	PSNR↑	SSIM↑	LPIPS↓
✗	✗	✗	✗	✗	22.53	0.759	0.240
✓	✗	✗	✗	✗	23.29	0.792	0.210
✓	✗	✓	✗	✗	23.70	0.801	0.207
✓	✓	✗	✗	✗	23.55	0.798	0.205
✓	✓	✓	✗	✗	24.67	0.835	0.169
✓	✓	✓	✓	✗	24.61	0.833	0.164
✓	✓	✓	✓	✓	**24.88**	**0.844**	**0.157**

Table 6. **Ablation Studies on DL3DV [27].** We quantify the benefits of geometric feature alignment, SH prediction, and depth correction after BA. For ablations, we use a smaller batch size of 32 for training and 6 inputs with a stride of 2 for evaluation.

Depth Shift with BA. Although subsequent BA using our aligned features improves poses, we find that directly substituting the new poses actually hinders NVS, as shown in the penultimate row of Tab. 6. This is because updating the camera parameters after refinement requires simultaneously moving all Gaussians to their correct locations and adjusting their scales accordingly. Our observations, explained in Sec. 3.3 and Fig. 4, suggest that this Gaussian transformation can be decomposed into a camera change and an affine transformation along the z-axis. By incorporating this adjustment, we achieve better NVS results, shown in the last row of Tab. 6.

5. Conclusion

We introduce Selfi, a self-improving approach for novel view synthesis from unposed and uncalibrated images. Our key insight is that while modern visual foundation models provide a strong prior for a variety of 3D reasoning tasks, their feature spaces may not be explicitly 3D aligned. Using VGGT as a VFM backbone, we train a feature adapter by constructing a self-supervised reprojection task, enforcing feature consistency between 2D projections that refer to a common 3D point. This geometric alignment not only improves downstream NVS performance but also yields robust correspondences for refining camera parameters via bundle adjustment. We can then feed the refined cameras back into the NVS model, and obtain further improvements in rendering quality. Using a self-supervised loop that aligns VFM features and refines poses, our model achieves state-of-the-art performance, demonstrating a powerful new direction for robust, unposed novel view synthesis.

Acknowledgment

We would like to thank Xichen Pan and Noah Snavely for insightful discussions during the project, and Clément Godard, Michael Broxton, and Maggie Oh for help with compute support. Additionally, we thank Stephen Lombardi, Ryan Overbeck, and Jason Lawrence for helpful suggestions and feedback.

References

[1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 2022. 7, 8

[2] Aleksei Bochkovskii, AmaÃĢl Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. In *ICLR*, 2025. 2

[3] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. 6

[4] Michael Broxton, John Flynn, Ryan Overbeck, Daniel Erickson, Peter Hedman, Matthew Duvall, Jason Dourgarian, Jay Busch, Matt Whalen, and Paul Debevec. Immersive light field video with a layered mesh representation. *ACM TOG*, 2020. 2

[5] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *CVPR*, 2024. 2, 6, 7

[6] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *ECCV*, 2024. 1, 2, 3, 6, 7

[7] Yue Chen, Xingyu Chen, Anpei Chen, Gerard Pons-Moll, and Yuliang Xiu. Feat2gs: Probing visual foundation models with gaussian splatting. In *CVPR*, 2025. 2

[8] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPRW*, 2018. 2

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words. In *ICLR*, 2021. 2

[10] Yilun Du, Cameron Smith, Ayush Tewari, and Vincent Sitzmann. Learning to render novel views from wide-baseline stereo pairs. In *CVPR*, 2023. 6, 7

[11] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world's imagery. In *CVPR*, 2016. 1, 2

[12] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. In *CVPR*, 2019. 1, 2, 3

[13] John Flynn, Michael Broxton, Lukas Murmann, Lucy Chai, Matthew DuVall, Clément Godard, Kathryn Heal, Srinivas Kaza, Stephen Lombardi, Xuan Luo, et al. Quark: Real-time, high-resolution, and general neural view synthesis. *ACM TOG*, 2024. 2, 3

[14] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022. 2, 4

[15] Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T. Freeman. Unsupervised semantic segmentation by distilling feature correspondences. In *ICLR*, 2022. 2

[16] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 2

[17] Tooba Imtiaz, Lucy Chai, Kathryn Heal, Xuan Luo, Jungyeon Park, Jennifer Dy, and John Flynn. Lvt: Large-scale scene reconstruction via local view transformers. In *SIGGRAPH Asia*, 2025. 2, 4

[18] Hanwen Jiang, Hao Tan, Peng Wang, Haian Jin, Yue Zhao, Sai Bi, Kai Zhang, Fujun Luan, Kalyan Sunkavalli, Qixing Huang, et al. Rayzer: A self-supervised large view synthesis model. In *ICCV*, 2025. 2, 7

[19] Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, et al. Anysplat: Feed-forward 3d gaussian splatting from unconstrained views. *ACM TOG*, 2025. 1, 2, 3, 4, 5, 6, 7

[20] Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. Lvsm: A large view synthesis model with minimal 3d inductive bias. In *ICLR*, 2025. 1, 2, 6, 7

[21] Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. Learning-based view synthesis for light field cameras. *ACM TOG*, 2016. 1

[22] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *ECCV*, 2024. 2, 7, 8

[23] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM TOG*, 2023. 1, 2, 3, 4, 6, 7, 8

[24] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM TOG*, 2017. 5, 7, 8

[25] Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3d with mast3r. In *ECCV*, 2024. 1, 2, 3

[26] Kai-En Lin, Lei Xiao, Feng Liu, Guowei Yang, and Ravi Ramamoorthi. Deep 3d mask volume for view synthesis of dynamic scenes. In *ICCV*, 2021. 2

[27] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *CVPR*, 2024. 5, 6, 7, 8

[28] Yifan Liu, Zhiyuan Min, Zhenwei Wang, Junta Wu, Tengfei Wang, Yixuan Yuan, Yawei Luo, and Chunchao Guo. Worldmirror: Universal 3d world reconstruction with any-prior prompting. *arXiv preprint arXiv:2510.10726*, 2025. 3, 4, 5, 6, 7, 8

[29] Grace Luo, Lisa Dunlap, Dong Huk Park, Aleksander Holynski, and Trevor Darrell. Diffusion hyperfeatures: Searching through time and space for semantic correspondence. In *NeurIPS*, 2023. 2

[30] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2021. 1, 2, 3

[31] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM TOG*, 2022. 2

[32] Riku Murai, Eric Dexheimer, and Andrew J Davison. Mast3r-slam: Real-time dense slam with 3d reconstruction priors. In *CVPR*, 2025. 2

[33] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *TMLR*, 2023. 2

[34] Linfei Pan, Daniel Barath, Marc Pollefeys, and Johannes Lutz Schönberger. Global Structure-from-Motion Revisited. In *ECCV*, 2024. 1, 2

[35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 2

[36] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021. 3, 5

[37] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 4

[38] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 2

[39] Paul-Edouard Sarlin, Ajaykumar Unagar, Mans Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, et al. Back to the feature: Learning robust camera localization from pixels to pose. In *CVPR*, 2021.

[40] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1

[41] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 1, 2

[42] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025. 2

[43] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Generalizable patch-based neural rendering. In *ECCV*, 2022. 2, 6, 7

[44] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, 2022. 2

[45] Stanislaw Szymanowicz, Chrisitian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. In *CVPR*, 2024. 1, 2

[46] Stanislaw Szymanowicz, Eldar Insafutdinov, Chuanxia Zheng, Dylan Campbell, Joao F Henriques, Christian Rupprecht, and Andrea Vedaldi. Flash3d: Feed-forward generalisable 3d scene reconstruction from a single image. In *3DV*, 2025.

[47] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *ECCV*, 2024. 2

[48] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. In *NeurIPS*, 2023. 2

[49] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *CVPR*, 2020. 2

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2

[51] Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. In *3DV*, 2024. 2

[52] Jianyuan Wang, Christian Rupprecht, and David Novotny. Posediffusion: Solving pose estimation via diffusion-aided bundle adjustment. In *ICCV*, 2023. 2

[53] Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Vggsfm: Visual geometry grounded deep structure from motion. In *CVPR*, 2024. 2

[54] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *CVPR*, 2025. 1, 2, 3, 8

[55] Peng Wang, Hao Tan, Sai Bi, Yinghao Xu, Fujun Luan, Kalyan Sunkavalli, Wenping Wang, Zexiang Xu, and Kai Zhang. Pf-lrm: Pose-free large reconstruction model for joint pose and shape prediction. In *ICLR*, 2024. 2

[56] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, 2021. 1, 2

[57] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *CVPR*, 2025. 2

[58] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *CVPR*, 2025.

[59] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024. 1, 2, 3

[60] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Scalable permutation-equivariant

visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025. 2

[61] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 2004. 7

[62] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. NeRF−−: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 5

[63] Haofei Xu, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Resplat: Learning recurrent gaussian splats. *arXiv preprint arXiv:2510.08575*, 2025. 1, 6, 7

[64] Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth. In *CVPR*, 2025. 2, 6, 7

[65] Linning Xu, Vasu Agrawal, William Laney, Tony Garcia, Aayush Bansal, Changil Kim, Samuel Rota Bulò, Lorenzo Porzi, Peter Kontschieder, Aljaž Božič, et al. Vr-nerf: High-fidelity virtualized walkable spaces. In *SIGGRAPH Asia*, 2023. 2

[66] Yinghao Xu, Zifan Shi, Wang Yifan, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetzstein. Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation. In *ECCV*, 2024. 1, 2

[67] Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In *CVPR*, 2025. 2

[68] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024. 2

[69] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *ECCV*, 2018. 2

[70] Botao Ye, Sifei Liu, Haofei Xu, Xueting Li, Marc Pollefeys, Ming-Hsuan Yang, and Songyou Peng. No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. In *ICLR*, 2025. 1, 2, 3, 4, 6, 7

[71] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021. 1, 2, 3, 6, 7

[72] Junyi Zhang, Charles Herrmann, Junhwa Hur, Luisa Polania Cabrera, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. A tale of two features: Stable diffusion complements dino for zero-shot semantic correspondence. In *NeurIPS*, 2023. 2

[73] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *ICLR*, 2025. 2

[74] Jason Y Zhang, Amy Lin, Moneish Kumar, Tzu-Hsuan Yang, Deva Ramanan, and Shubham Tulsiani. Cameras as rays: Pose estimation via ray diffusion. In *ICLR*, 2024. 2

[75] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. In *ECCV*, 2024. 2, 6, 7

[76] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 7

[77] Shangzhan Zhang, Jianyuan Wang, Yinghao Xu, Nan Xue, Christian Rupprecht, Xiaowei Zhou, Yujun Shen, and Gordon Wetzstein. Flare: Feed-forward geometry, appearance and camera estimation from uncalibrated sparse views. In *CVPR*, 2025. 2, 6, 7

[78] Qitao Zhao, Amy Lin, Jeff Tan, Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. Diffusionsfm: Predicting structure and motion via ray origin and endpoint diffusion. In *CVPR*, 2025. 2

[79] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM TOG*, 2018. 1, 2, 5, 6, 7

[80] Chen Ziwen, Hao Tan, Kai Zhang, Sai Bi, Fujun Luan, Yicong Hong, Li Fuxin, and Zexiang Xu. Long-lrm: Long-sequence large reconstruction model for wide-coverage gaussian splats. In *ICCV*, 2025. 1, 2, 6, 7